

== X1B Pack v1.0 ==

The **x1b pack** is a series of perl scripts ive written and/or rewritten over the time to help me learn and better understand Perl Sockets Programming. My sources for learning consisted of the “Perl Little Black Book” <http://www.amazon.com/exec/obidos/tg/detail/-/1576104265/103-2472564-0287027?v=glance> series, CPAN <http://www.cpan.org> , Perl Doc <http://www.perldoc.com> and O’reillys “The Perl CD Bookshelf” (the camel is your friend) <http://www.oreilly.com/catalog/perlcdb2> . My main stepping stone to learning perl has always been through learning sockets first as it kept my interest, then research and learn other features such as regular expressions as the occasion arose to learn and understand them.

In this pack you should find:

- **x1bscan** (a port & service connect() scanner /w banner grabbing)

This scanner is far from fast but is accurate and stable. Not suggested for production use but rather for a stepping stone to learning perl sockets. I took the time to study each individual port as provided in the IANA ports list <http://www.iana.org/assignments/port-numbers> and research the definitions of the abbreviated ports then provide the full definition in my ports @array as opposed to leaving the user guessing as to just what the located port was for.

- **x1bdns** (a DNS service enumeration script /w zonetransfer)

During my research of port 53 DNS I learnt what information could be gleamed from a DNS service that allows communication via the UDP protocol. Taking this knowledge and researching the Net::DNS resolver module I wrote this script to teach my self hands-on what exactly was happening and why.

- **x1btelnet** (a telnet/router remote bruteforce login/password cracker)

A bit of a mess and not sure it still works but this was my first step into figuring out the Socket module along with arrays, scalar variables, hashes, File IO and handles. This script led to some interesting research on the telnet service and router insecurities. I’ve provided a list of known default login and passwords for assorted router vendors for your use.

- **x1bweb** (a cgi web script scanner)

Fairly vanilla web script scanner. No anti ids or stealth features and doesn’t check response codes <http://kbs.cs.tu-berlin.de/~jutta/ht/responses.html> for false positive filtering, but is fast. Really not suggested for production use. This script was inspired by my first cgi script scanner that checked for the whois_raw.cgi script and vulnerability which infact was ripped from piffys issscan.pl scanner that I hacked to mass scan for the whois_raw.cgi script and attempt to exploit it. Last I checked it was available at <http://www2.packetstormsecurity.org/cgi-bin/search/search.cgi?searchvalue=whois+raw&type=archives> ...which now that I look at it again after 2 years, is kind of embarrassing...but such is the learning curve. It really is in poor taste to rip code and not at least give credit to the original author. Just don’t do it.

- **x1bpop3** (a pop3 remote login/password bruteforce script /w SSL support)

This script was my first step into SSL sockets programming and is a hack from my x1bftp script with the Net::FTP module swapped for the Mail::POP3Client module and a few adjustments made to fuction with POP3. Not really rocket science but was fun none the less.

- **x1bftp** (an ftp remote login/password bruteforce script)

x1bftp was an excuse to abuse and practice regex’s (regular expressions) more while I was reading the Activestate “perlretut” that came with activestates perl user guide. While you’re thinking to your self “oh! X1b writes code on windows! So should I!” No. Actually sockets is far more convienient from a UNIX/Linux as such modules as Net::RawIP and Net::Pcap are either better supported on a *NIX or just not supported on a Windows. Though I do not bash windows. When I start writing my own OS’s ? Maybe then I’ll grow a justified ego and starting judging windows.

- **x1bgetraw** (an http banner query script)

x1bgetraw was my answer to the “telnet www.domain.com:80” method of grabbing webserver banner information, perl style. As the perl moto goes, “there is more than one way to do it”. X1bgetraw allowed me to grab banner information, time stamps, OS info, web script info, cookie info in a much cleaner output format than the telnet method. I actually ment to and likely later will add the option to try varied method calls besides just “HEAD”. <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html> . It’s interesting to see the different reactions webserver give to different method calls.

- **x1bping** (multi ping script)

My wife bought me the Java2 core language Little Black Book not long ago. As is my routine I immediately opened the book’s index and searched for “sockets”. Sure enough, Chapter 8, page 209 there is a complete chapter on Java Sockets Programming. Under “Essential Socket Programming” paragraph 2 I read “For example, java sockets support User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) connections. If you want something else – for example, Internet Control Protocol (ICMP) for a ping program – you’ll have to resort to native method calls (probably written in C)”. At which point I made a soure face and opened Mozilla, went to CPAN and searched for Net::Ping, copied the example source and began building around it. Minutes later I have x1bping and relieved that perl wont make me resort to a 2nd language to complete a task. God bless Perl. Don’t get me wrong, I’ve seen some very impressive Java Sockets Programs such as blackhat.be’s <http://www.blackhat.be/cst/> but suddenly I’m in no rush to explore java sockets and am content to continue my perl studies.

Also in **X1B Pack** V1.0 you should find a compiled windows binary of each source for the perl impaired and our loving script kiddies. The wordlist and loginlist are for use with my login/password bruteforce scripts. These list are the compilation results of bruteforce attack and password cracking research. They aren’t in any specific order. You will need to create your own hostlist files and log files for use with the x1b scripts.

If for any reason my work should interest you, I have additional work scattered across the net that seems to have propagated it’s self from www.packetstormsecurity.org under my hobby name “NeoErudition Technologies” and true name Lawrence LaVigne. X1b is only used to label my hobby code from my production code. X1b being a perl ESC code. Perl being my Escape.